A Hardware-based Smart Camera for recovering High Dynamic Range video from multiple exposures

Pierre-Jean Lapray[a], **Barthélémy Heyrman**[a] **and Dominique Ginhac**[a]
[a]University of Burgundy, Le2i UMR 6306, Dijon, France

Abstract. In many applications such as video surveillance or defect detection, the perception of information related to a scene is limited in areas with strong contrasts. The high dynamic range (HDR) capture technique can deal with these limitations. The proposed method has the advantage of automatically selecting multiple exposure times to make outputs more visible than fixed exposure ones. A real-time hardware implementation of HDR technique that shows more details both in dark and bright areas of a scene is an important line of research. For this purpose, we built a dedicated smart camera which performs both capturing and HDR video processing from three exposures. What is new in our work is shown through the following points: HDR video capture through an Multiple Exposure Control, HDR memory management, HDR frame generation and representation under a hardware context. Our camera achieves a realtime HDR video output at 60 fps at 1.3 MegaPixels and demonstrates the efficiency of our technique through an experimental result. Applications of this HDR smart camera include the movie industry, the mass-consumer market, military, automotive, and surveillance.

Keywords: image reconstruction, image enhancement, imaging systems, video.

Address all correspondence to: Dominique Ginhac, University of Burgundy, Le2i UMR 6306, Dijon, France, 21000; E-mail: dginhac@u-bourgogne.fr

1 Introduction

Standard cameras have a limited dynamic range. In many video and imaging systems, we have saturated zones in the dark and illuminated areas of the captured image. These limitations are due to large variations of the scene radiance, with over and under-exposed areas in the single image. The sequential capture of several images with different exposure times can deal with the lack of information in extreme lightning conditions.

According to Krawczyk et al.,[1] there are two types of devices that can be used to capture the entire dynamic of a scene: the HDR sensors and standard sensors. The HDR sensors are, by design, able to capture a wide dynamic range with a single capture. However, these sensors are still under development and are not suitable for embedded and low cost applications. Another technique is to use a standard LDR (Low Dynamic Range) sensor. This technique derives several ways to proceed:

- changing the exposure time by multiple captures which is the most common method ;
- spatial exposition: capture simple, with a mask in front of the sensor ;
- multiple sensors with a shared light beam.

According to A. E. Gamal,[2] the multiple capture technique is the most efficient method, widely used in recent works.[3–5] Create an HDR image is done in three steps:

- recover the response curve of the system ;
- blend pixels into radiance values ;
- perform Tone Mapping to match the dynamic range of the scene to that of the display device.

This technique is designed to calculate the light intensities of real scenes, where each pixel is stored on a very large dynamic range (up to 32-bits wide and more). It is therefore necessary to have a large range of memory to store images, and to reconstruct the HDR image.

The three most popular algorithms for HDR reconstruction are those of Debevec and Malik[6] , Mitsunaga et al.[7] and Robertson et al.[8] A technical paper by Yourganov[9] compares the first two algorithms implemented in C/C++ in real time on PC. The results of computation time for an image are substantially the same. Mitsunaga algorithm calculates the benefit of the response curve without knowing the exposure times of the different images to merge. They perform automatic rejection of the image parts with significant effects of vignetting, or temporal variations. Originally, the Debevec method has been developed for photography. However, according to the research conducted by Yourganov,[9] this method can be easily applied to digital video, both for static and dynamic scenes, if captures are fast enough that light changes between two consecutive frames

can be safely ignored. Consequently, such a method is widely used to produce HDR video, by capturing frames with alternating bright and dark exposures, as pointed by Tocci[10]

HDR creating is followed by the tone mapping operation.[11] It is used to render the HDR data to match the dynamic range of conventional hardware displays. For example, it converts 32-bit wide pixels to 8-bit wide pixels ([0,255]). There are two types of tone mapping operators (TMO): spatially uniform (global TMO) and spatially non-uniform (local TMO). In our case, several algorithms seem to be implementable in real-time due to fast computation capabilities, whether global or local. Following is a list of methods known for their efficiency and simplicity. These methods are ordered from the fastest algorithm to the slowest algorithm implemented in software:

- Durand et al.[12] (*Fast Bilateral Filtering for the Display of High-Dynamic-Range Images*) ;
- Duan et al.[13] (*Tone-mapping high dynamic range images by histogram novel Adjustment*) ;
- Fattal et al.[14] (*Gradient Domain High Dynamic Range Compression*) ;
- Reinhard et al.[15] (*Photographic Tone Reproduction for Digital Images*) ;
- Tumblin et al.[16] (*Time-dependent visual adaptation for fast realistic image display*) ;
- Drago et al.[17] (*Adaptive Logarithmic Mapping For Displaying High Contrast Scenes*).

Representing as closely as possible the reality is a crucial aspect to be considered for any HDR system. Several publications have been made concerning the evaluation of tone mapping methods.[18] A study by Akyz et al.[19] shows that the method by Reinhard et al.[15] photographic tone mapping gives the best results in terms of natural appearance in the image. Another study by Drago et al.[20] reaches the same conclusion. Moreover, Cadik[21] shows very good results for the

operator to Reinhard. There is a surprising aspect in this article: the global part of the methods of tone mapping is critical to achieve good visual results for real-world scenes.

In this paper, we propose a HDR smart camera based on a parallel hardware architecture dedicated to the production of real-time HDR video content from a set of different exposures. From an end-user point of view, the HDR video must be built and delivered at full resolution and at the sensor framerate with no detectable latency from each new capture and frames previously stored. Moreover, this HDR platform embeds all the necessary algorithms to automatically evaluate the best exposure times from any visual scene in order to provide the best HDR content. What is new in this paper is shown through four steps. First, we capture images from the sensor with alternating three exposure times, selected automatically by our Multiple Exposure Control (MEC). Then, we manage reading and writing operations in memory in order to have several video streams in parallel, corresponding to the different exposure times. Under a highly parallel context, we blend the three video streams together with a modified version of a standard HDR technique. Finally, an hardware implementation of a global tone mapping technique is performed. We will begin this paper by describing existing works about HDR video technique in Section 2. Then, in Section 3, we will describe our system in detail. Finally, some experiments and results will follow this discussion in Section 4. Concluding remarks are then presented.

2 Related work

We detail here the existing hardware architectures. We limit ourselves to recent systems which can operate in real-time, whether they are focused exclusively on capture, HDR creating or tone mapping. Table 1 summarizes these methods.

In 2012, Akyüz et al.[28] developed a complete system on a GPU (”Graphics Processing Unit”)

Table 1 Summary of the main embedded real-time architectures dedicated to HDR.

Method	Hardware	Capture	Fusion HDR	Frames used	Tone Mapping	Resolution	FPS
Akyüz et al.[22]	GPU	no	yes	9	yes	-	65
Mann et al.[23]	FPGA	yes	yes	3	yes	$1,280 \times 720$	120
Ureña et al.[24]	GPU/FPGA	no	no	-	yes	640×480	30/60
Guthier et al.[25]	CPU+GPU	yes	no	-	no	640×480	25
Ching-Te et al.[26]	ARM SOC	no	no	3	yes	$1,024 \times 768$	60
Bachoo et al.[27]	CPU+GPU	no	yes	3	-	$1,600 \times 1,200$	20

platform. The tasks are performed in parallel with a pipelined structure. Generating HDR and the tone mapping are done without knowing the response curve of the camera. They use the algorithm originally proposed by Debevec et al.[6] to estimate the radiance values. Regarding to the operation of the tone mapping, the Reinhard et al.[15] algorithm has been chosen and implemented. Some results are identical compared to other methods implemented on CPU. They reach a framerate of 65 fps for producing HDR images, and 103 frames per second for performing the tone mapping. However, they do not have time to load textures on the GPU. The majority of time is spent in sending pixels to the GPU. Radiance computations and weighting have little impact on the speed calculation, and the framerate of the final system.

The most popular complete HDR vision project is based on the Mann[23] system. In 2012, a welding helmet composed of two computer-controlled video cameras has been presented. The data received by these cameras are recorded line by line in an external memory. Several FIFOs store the pixels and read them simultaneously line by line. The number of FIFOs depends on the number of images used by the HDR reconstruction module. A LUT containing precomputed values is used to combine multiple exposures. This LUT is inspired of the work by Ali et al.[29] , the estimation of

radiances is done with a CCRF (”Comparametric Camera Response Function”). With this method, they are able to obtain a video with a fixed latency, and a controlled time calculation on a Xilinx Spartan-6 LX45 FPGA.

Ureña et al.[24] published in 2012 two tone mapping architectures, described both on GPU and FPGA. The implementations were done on a battery portable operating circuit. A new generation of tone mapping is presented in this article. The tone mapping operator includes both local and global calculation. Typically, for the overall look, it highlights areas containing low contrasts, but can also protect areas where the contrast is well. Locally, it reduces the areas that are too bright in order to improve the image details. The overall improvement is based on the brightness histogram adaptation of each channel in the HSV colour space. On the other hand, the local enhancement is based on the retina-like technique. To summarize, the Gaussian filters, the weighting and the human visual system consideration are the main advantages of the operator. The FPGA implementation produced a video with a high frame rate, consuming little electric power, while the GPU implementation provides greater precision in the calculation of HDR pixels, but uses a lot of resources.

In 2012 Guthier et al.[30] introduced an algorithm with a good HDR quality, that can be implemented with the same number of LDR (Low Dynamic Range) captures. The choice of exposures is performed optimally by selecting the better shutter speeds that will add the more useful information to contribute to the final HDR image. The context can be real-time, by minimizing the number of images. Basically, the exposure times are wisely chosen so that at least one LDR image has a well exposed pixel at one position (i, j). First, a good approximation of the radiance value E is calculated taking into account the response function of the camera and a contributing function. A useful relationship is made between the radiance histogram vector and the contribution of each

images that indicate potentially changes in the scene. A stability criterion is also introduced to the sequence which allows each frame to be adjusted until a stable shutter sequence is found. Finally, with this algorithm, it saves capturing time and reduces the number of LDR exposures without loss of quality at the end of the computation.

Ching-Te et al.[26] suggests a methodology to develop a tone mapping processor optimized using an ARM SOC platform (System On Chip). Their processor evaluates both photographic compression method by Reinhard et al.,[15] and the gradient compression method by Fattal et al.,[14] for different applications. The new processor can compress $1,024 \times 768$ HDR images at 60 fps. The core needs $8,1mm^2$ of physical area with $0.13m$ TSMC technology.

Bachoo[27] developed a dedicated technical application of exposure fusion (initiated by Mertens et al.[31]), to merge a real-time 1600×1200 video at 20 fps using three black and white videos. They are able to control the speed of image generation, to have a constant frame rate, relative to the defined processing block size. They perform an alternative Goshtasby[32] algorithm. The implementation is done on CPU and GPU. The algorithm is divided into two parts so that the power of the CPU processing and GPU is used wisely. The CPU perform massively sequential operations such as calculating entropy blocks. The GPU is used to merge the blocks together, operation which can be parallelized to increase execution speed of the fusing process. The speed can be increased if the video resolution is reduced or if the size of processing blocks increases. As this, a compromise between calculation speed and quality can be chosen. Nothing is said about the choice of exposure time and no method is proposed to estimate exposures. It is recorded that the use of additional exposures may produce a bottleneck in the fusing process.

3 A dedicated HDR Smart Camera

The dedicated hardware platform is a smart camera built around a Xilinx ML605 board, equipped with a Xilinx Virtex-6 XC6VLX240T (see Figure 1(a)). The motherboard includes a 512 MB DDR3 SDRAM memory used to buffer the multiple frames captured by the sensor. Several industry-standard peripheral interfaces are also provided to connect the system to the external world. Among these interfaces, our vision system implements a DVI controller to display the HDR video on an LCD monitor. It also implements an Ethernet controller to store frames on a host computer. A custom-made PCB extension board has been designed and plugged into the FPGA board to support the Ev76c560 image sensor, a 1280 x 1024-pixel CMOS sensor from e2v. It offers a 10-bit digital readout speed at 60 fps in full resolution. It also embeds some basic image processing functions such as image histograms, evaluation of the number of low and high saturated pixels. Each frame can be delivered with results of these functions encoded in the video data stream header

Insert Figure 1 here

Fig 1 Overview of our HDR smart camera.

The parallel architecture presented in this paper operates in several stages. At the first stage, an FPGA input interface receives sequentially three pixel streams (produced by the e2v sensor EV76C560), and stores them to a memory as mosaiced colour images. No demosaicing is performed at this time. A Multiple Exposure Control (MEC) based on the histogram computation also operates in parallel to select the proper exposure times. It changes the sensor configuration each time an image is captured. The second stage is based on a memory management core which

reads the previous frames stored into the SDRAM, and delivers it as a synchronized parallel video outputs. At the third stage, the different pixel streams are combined into an HDR frame, knowing the response curve of the imaging system and the exposure times of images. This stage produces a complete radiance map of the captured scene. Finally, the High Dynamic Range frame is tone mapped and can be displayed on a standard LCD monitor via a DVI controller. This full process is continuously updated in order to perform a real-time HDR live video at 60 fps with a 1280×1024 pixel resolution. Our real-time constraint is that the availability of a new HDR data from the LDR captures must not exceed a fixed latency of $1ms$, guaranteeing that the HDR process is imperceptible to the viewer.

Our camera must be automatically adapted to the illumination level, just as the human eyes do. The best set of exposures has to be evaluated in order to capture the adequate dynamic range of the scene. But, when we perform HDR stitching, traditional auto exposure algorithms fail. We present a similar approach of a previous state of the art algorithm by Gelfand et al.[33] , adapted to our real-time hardware requirement. Our sensor is able to send us the complete image histogram. Using the histogram of each image will allow to have a preview of the total range of brightness that is being recorded. We require that fewer than 10% of the pixels are saturated in white for the short exposure, and require that fewer than 10% of pixels are saturated in dark for the long exposure, as illustrated in Fig. 2.

Insert Figure 2 here

Fig 2 From left to right, the first image shows 15.3% of pixels saturated at high level for the short exposure before MEC, whereas the second image has only 5.4% of these pixels after MEC. In the same manner, the third image shows 38.3% of pixels saturated in black before MEC and the fourth image only includes 0.2%. after MEC.

However, the method developed by Gelfand for the evaluation of the exposure times has been

specifically designed for HDR photography on smartphone. Their approach is optimal to capture a single HDR image but cannot be considered for an HDR video. In our approach, we decide to continuously update the set of exposure times from frame to frame to minimize the number of saturated pixels by instantaneously handling any change of the light conditions. The estimation of the best exposure times is computed from the 64-level histogram provided automatically by the sensor in the data-stream header of each frame. Let's call Δt_L, Δt_M and Δt_H the three exposure times related to our scene. Image histograms provided by the sensor are encoded with 64 categories, with 16 bits by category. According to the captured images with low exposure I_L and high exposure I_H times, we apply these functions:

$$Q_L = \sum_{h=1}^{h=4} \frac{q(h)}{N} \quad Q_H = \sum_{h=60}^{h=64} \frac{q(h)}{N} \,, \tag{1}$$

where Q_L and Q_H are the proportion of pixels on a specific part of the histogram, among N pixels that compose an image. q_h is the number of pixels in each bin h. The calculation is done with the first four and the last four categories in the images I_H or I_L. The output pixels are encoded with 10-bit (between 0 and 1023), four categories correspond to a range of 64 pixel values. Then we calculate one parameter for both extreme exposure times like this:

$$\delta Q_{L/H} \quad = \quad |Q_{L/H} - Q_{L/H,req}| \,, \tag{2}$$

where $Q_{L/H,req}$ is the required pixel quantity for a specific part of the histogram (10% among N). $\delta Q_{L/H}$ evaluates how far is the amount of current pixels with the desired quantity. Once we

have these parameters, the system takes a series of decisions to the next image captures at $t+1$:

$$\Delta t_{L/M/H,t+1} \quad \leftarrow \quad MEC(\Delta t_{L/H,t})\,, \tag{3}$$

$$\Delta t_{L,t+1} = \begin{cases} \Delta t_{L,t} \pm 1x \text{ for } \delta Q_L > thr_{Lm} \\ \\ \Delta t_{L,t} \pm 10x \text{ for } \delta Q_L > thr_{Lp} \end{cases} \tag{4}$$

$$\Delta t_{H,t+1} = \begin{cases} \Delta t_{H,t} \pm 1x \text{ for } \delta Q_H > thr_{Hm} \\ \\ \Delta t_{H,t} \pm 10x \text{ for } \delta Q_H > thr_{Hp} \end{cases} \tag{5}$$

$$\Delta t_{M,t+1} = \sqrt{\Delta t_{L,t} \Delta t_{H,t}}\,, \tag{6}$$

where $\Delta t_{L/M/H,I}$ are the values of exposure time of the current images I_L, I_M and I_H.

To obtain a correct convergence time, the exposure time is automatically adjusted using two different levels of thresholds: one for a small variation of illumination, and one for greater range of illumination changes. thr_m (minus) and thr_p (plus) are the two threshold values that correspond to two different levels of action on the adjustment of exposure time. These thresholds will directly affect the transition speed to a stable state. For example, when you have a sudden increase in light in a short space of time, we affect the sensor exposure time according to thr_p. x determines how we change exposures. Here it corresponds to the sensor time line, $x = 15.72us$.

4 Implementation

4.1 Specific HDR Memory Management Core

The use of external off-chip memories is judicious for our application that processes large amount of data and high data rates. For our case, video processing requires two frames of data to be stored. In practice, this storage is implemented using DDR3 SDRAM chip which is a part of our hardware development platform. It requires fast and efficient direct memory access logic to achieve high dynamic range video in real-time.

Insert Figure 3 here

Fig 3 Memory Management Core Initialization. The sensor sends sequentially low (I_1) and middle (I_2) exposure times. Writing operations into memory of each rows Λ indexed by λ of the first two frames.

Insert Figure 4 here

Fig 4 Memory Management Core. Performing three parallel streaming videos with low (I_1), middle (I_2) and high (I_3) exposure times. The delayed HDR row output is shown after HDR and tone mapping computations (related to Section 4.2 and 4.3).

The sensor is able to send full-resolution images at 60 frames/s. Initialization of our specific HDR Memory Management Core is shown in Fig. 3. I_1 and I_2 are first stored in DDR3 memory. The first frame (I_1) is stored row by row with the function $W\Lambda_\lambda I_1$, where λ indexes row number ($1 <= \lambda <= 1024$). For example $W\Lambda_1 I_1$ means "writing of the first row $\Lambda 1$ of I_1 into memory". Each row write operation is followed by inter-row delay, due to horizontal sensor synchronization. For the second frame I_2, the image is also stored row by row ($W\Lambda_\lambda I_2$). This initialization step is required before the generation of the first HDR frame. We can't avoid waiting for these two first exposures. After this step, the Memory Management Core can start (see Fig. 4).

During the capture of the last frame (I_3), rows of the two previous frames stored are synchronously read from the memory during inter-frame ($R\Lambda_\lambda I_1$, $R\Lambda_\lambda I_2$) and buffered into Block RAMs (BRAMs) while each new captured row ($W\Lambda_\lambda I_3$) is stored in memory. It's important to notice that the design is a pure-hardware system which is processor-free and must be able to absorb a continuous pixel flow of about 80 MegaPixels per second from the sensor (called "Memory In" in Fig. 3 and in Fig. 4) while reading two other pixel flows corresponding to the two stored images (respectively called "Memory Out 1" and "Memory Out 2" in Fig. 4).

The HDR content is computed with the methods described in Sections 4.2 and 4.3. The HDR process needs a continuous stream of pixels of three images and then can only be performed while receiving the third frame I_3. Then, the process can iterate throughout the capture of the fourth frame (low exposure I_4) and the readout of the second and third frame (I_5 and I_6). Finally, our memory management system is able to deliver two parallel pixel streams that have been acquired and stored into the memory and a third pixel stream directly from the sensor. With this technique, each HDR pixel only requires three memory accesses (one write and two read operations during one row interval), saving many memory access operations. The main advantages of such a technique are (1) to store only two images in memory, and (2) to avoid the waiting for the three images to compute an HDR image. A latency corresponding to 136 clock rising-edges (i.e. $1.2us$ for a $114MHz$ system clock) is required by the system to create HDR tone mapped data (grey part of HDR output in Fig. 4) from the three captured lines. And then, it delivers an HDR video stream at 60 fps directly updated at each time the sensor sends an image.

4.2 HDR creating

The evaluation of the response curve of the system g only requires the evaluation of a finite number of values between Z_{min} and Z_{max} (typically 1,024 values for a 10-bit precision sensor), as depicted in the paper of Debevec and Malik.[6] This evaluation is not required if the camera has a linear response. However, for the major parts of image sensors, including the sensor used in our hardware platform, the response is not linear. The most significant nonlinearity in the response curve is around the saturation points (i.e. very dark pixels and very bright pixels), where any dark (respectively bright) pixel with a radiance below (respectively above) a certain level is mapped to the to the same minimum (respectively maximum) image value. The evaluation of the response curve has not been implemented on the hardware platform because it needs to be computed only once for a given sensor. So, these values are preliminarily evaluated by a dedicated PC software (Matlab code provided with the Debevec paper) from a sequence of representative images, and then stored into a Look-Up Table (LUT, 1,024-word memory) on the FPGA, for further reuse to convert pixel values. For recovering the HDR luminance value E_{ij} of a particular pixel, all the available exposures of this pixel are combined using the following equation:

$$\ln E_{ij} = \frac{\sum_{p=1}^{p=3} \omega(Z_{p,ij})[g(Z_{p,ij}) - \ln \Delta t_p]}{\sum_{p=1}^{p=3} \omega(Z_{p,ij})} \,, \tag{7}$$

where p indexes image number, i and j indexes pixel position, Δt is the exposure time and $\omega(z)$ is a weighting function giving higher weight to values closer to the middle of the function:

$$\omega(z) = \begin{cases} z - Z_{min} \text{ for } z \leq \frac{1}{2}(Z_{min} + Z_{max}) \\ \\ Z_{max} - z \text{ for } z > \frac{1}{2}(Z_{min} + Z_{max}) \end{cases} \tag{8}$$

where Z_{min} and Z_{max} values depend on the sensor output dynamic (typically $1,024$ values for a 10-bit precision sensor).

Insert Figure 5 here

Fig 5 HDR creating and tone mapping hardware pipeline using three different pixel streams. Frame enable is active when a new HDR frame is coming. It is important to note that we use IEEE754 32-bit floating-point arithmetic operators.

Considering $Z_{1,ij}$, $Z_{2,ij}$ and $Z_{3,ij}$ as $Z_{L,ij}$, $Z_{M,ij}$ and $Z_{H,ij}$, the overall scheme is visible in a pipeline architecture in Fig. 5. Computation of luminance values requires the use of 32-bit arithmetic operators (subtractors, multipliers etc.) and transition from 10-bit to IEEE754 32-bit wide (called "Fixed-to-Float" in Fig. 5). LUTs are used to store response curve g and make the transition from exposure time values $\Delta t_{L/M/H}$ to neperian logarithm field. These LUTs are used to avoid too much large hardware utilization. Floating-point operators with a large data bus have been chosen according to a detailed study on arithmetic operators on FPGA,[34] focusing on estimation surface and time for floating and fixed operators. They note that the surface increases exponentially with the accuracy (number of bits of representation). In addition, output delays increase linearly with precision. In view of these results, it was particularly interesting to consider implementations provided by Xilinx, including floating-point operators. The choice of using floating algorithms became spontaneously, given the huge dynamic computations for radiances. Moreover, as indicated in Table 2, floating architecture does not consume significantly more resources than the fixed-point architecture.

Operator	Fixed point		Floating point	
	LUTs	DSPs	LUTs	DSPs
Add/Sub	75	0	477	0
	0	1	287	2
Multiplicator	696	0	659	0
	132	1	107	3
Dividor 1 Cycle	1377 (Radix-2)	0	780	0
Dividor 25 Cycles	-	-	187	0
Root mean square 1 Cycle	1550	0	533	1
Root mean square 25 Cycles	-	-	170	1

Table 2 Resource comparison of fixed and floating point arithmetic operators on Virtex 6

4.3 Tone Mapping

Once the radiance map is recovered, image pixels have to be mapped to the display range of a selected material. In our case, the displayable range is 2^8 values. Reinhard et al.[15] require one global computation: the log average luminance found in the image, calculated as

$$\bar{E}_{ij} = \exp\left(\frac{1}{N}\sum_{i,j}\ln E_{ij}\right), \tag{9}$$

where E_{ij} is the scene radiance for pixel (i,j), N is the total number of pixels in the image. Then, we want to map the middle-gray scene luminance to the middle-gray of the displayable image. For the photographic tone reproduction operator, an approach is to scale the input data such that the log average luminance is mapped to the estimated key of the scene:

$$\begin{aligned} D_{ij} &= 255 \cdot \frac{a\frac{E_{ij}}{\bar{E}_{ij}}}{1+a\frac{E_{ij}}{\bar{E}_{ij}}} \\ &= 255 \cdot \frac{1}{1+\frac{\bar{E}_{ij}}{a\cdot E_{ij}}}, \end{aligned} \tag{10}$$

where a is a scaling constant appropriate to the illumination range of the image scene. We chose 0.18 empirically in our case.

5 Results and discussion

5.1 Hardware implementation

Our work has been implemented on a Virtex-6 platform. We show the hardware implementation results in Table 3. Usually, FPGA-based image processing requires many specific devices such as SRAM memory, multi-port memory, video direct memory access, dedicated processors, and consequently, consumes many DSP blocks. This is not the case for our implementation. It consumes relatively low hardware complexity since the number of occupied slices is $6,692$ (about 17% of the device) and the number of LUTs is $16,880$ (i.e. 11% of the device). These results highlight several interesting points. First of all, since the hardware utilization is limited with the Virtex-6 platform, it let us the opportunity to implement the full HDR pipeline onto a less powerful FPGA like a Xilinx Spartan-6 LX45, providing a more low-cost HDR smart camera. Secondly, our technique can be extended to more complex processing architectures. Among them, we can cite more complex tone mapping operators and specifically local operators, known to give enhanced visual performance. We can also mention HDR pipeline using more than three LDR exposures in order to capture more details in the scene.

Two series of captures of digital still images from the different video LDR streams are shown in Fig. 6. For the two sets, you can see from left to right the contributions from the different LDRs frames (low, medium and high exposures) and the HDR image. As an example, for the first series, we can distinguish the word ”HDR” inside the lamp (high brightness), and the word ”HDR” inside the tube (low brightness).

Table 3 Summary of hardware implementation results on the Virtex-6 platform.

Metric	Utilization/Availability	%
Estimated supply power	6.039 W	
Maximum frequency	126.733 MHz	
Number of occupied Slices	6,692 out of 37,680	17%
LUTs	16,880 out of 150,720	11%
Registers	20,192 out of 301,440	6%
Number of bonded IOBs	196 out of 600	32%
36K BRAMs	17	4%

Insert Figure 6 here

Fig 6 Results of the complete system. Our Multiple Exposure Control can select the three proper exposures, and the specific memory management core permits us to display the 3 bracketed images. The HDR image is in the right of each image set.

Our design has an horizontal blanking period of 307 pixels, and a vertical blanking period of 20 rows. The entire design contains hardware and algorithm latencies. The efficient latency at the end of each row is 127 extra-clock ticks (whether $1.11\mu s$ for a clock pixel of $114MHz$). This constant latency appears but not alters the frame rate. Indeed, we use the horizontal blanking periods delivered by our sensor to compensate the latency. With a video frame rate of 60 frames per second, our system is able to process $60 \times (1280 + 307) \times (1024 + 20) = 99.40$ Mega pixels per second. The hardware system has a maximum operating frequency of 126.733 MHz. Since the video input and video output interfaces are running at 114 MHz, is it acceptable to have a lower system clock? The answer is yes because during active video, we will support the back-pressure

Insert Figure 7 here

Video 7 Output video (MPEG, 3 MB).

on the slower clock using the BRAMs. During blanking periods, the BRAMs will empty, and the interface will catch up. Finally, our architecture embeds all the algorithmic operators to produce a single tone-mapped output pixel in real-time.

5.2 Visual quality

HDR image quality metrics require the availability of a reference image with which the images using the different tone mapping operators is to be compared. In this paper, the reference image is the Paul Debevec's HDR photo of Stanford memorial church used historically in the seminal works on HDR.[6] This reference image has been built from the original LDR exposures series of the church, and tone-mapped with the adaptive logarithmic mapping of Drago et al.[17] This technique is described as the most natural method and also the most detailed method in dark region.[35] A specific test software implementing the different tone mapping operators has been developed in Matlab and used to evaluate precisely the quality of each tone mapper. Since our hardware implementation relies on 3 exposures, we used the three LDR images of Figure 8, in order to compare the different methods. The exposure time for these three images are respectively 32s, 1s and 31ms from left to right.

Insert Figure 8 here

Fig 8 The three low dynamic range images used for quality comparison.

Image quality metrics can be divided into two main categories. The first category are difference based metrics. Among them, the most widely used quality metrics are mean square error (MSE) and peak signal-to-noise ratio (PSNR) because they are simple mathematical measures evaluating the distortion between the image and the reference. However, they are not well matched to per-

ceived visual quality because they do not take the characteristics of the human visual system into account.[36] For the performance evaluation of different tone mapping operators, subjective criteria as the human perception is of crucial importance. So, the second category of quality metrics include only human visual system based metrics. Among them, Universal Quality Index (UQI[37]) and Structural SIMilarity (SSIM[36]) are used for measuring the perceptual similarity of the tone mapped images. UQI is an image quality index that models the image distortion as a combination of three factors: loss of correlation, luminance distortion, and contrast distortion. In UQI, local statistics are computed to estimate a similarity between all corresponding 8×8 blocks across input and reference images. The SSIM index is a generalized form of UQI for measuring the similarity between two images. In the SSIM metric, the image structure is represented by statistical measures (mean and variance), and image quality is measured based on the similarity between the structure of the reference and the test image. So, a high-quality test image has a structure that closely matches the structure of the reference. SSIM is based on a specific measure of spatial correlation between the structure of the images to quantify the degradation of the image structure.[38] This metric imitates the human perception on image structure and returns results that are more consistent with the human visual system than MSE and PSNR.

Table 4 summarizes the comparison results in terms of image quality of the tone mapped images produced by our technique and by other methods using the above mentioned metrics: Universal Quality Index (UQI), Structural SIMilarity (SSIM), mean square error (MSE), normalized root mean square error (NRMSE), and computation times. In terms of visual quality, our method outclasses all the other methods (Drago et al., Schlick et al., and Tumblin et al.) with the highest values both for UQI (0.89) and SSIM (0.8), while having similar computation times. Moreover, our method gives better performance than the high-complexity local method proposed by Rein-

hard et al. because we obtain identical performance in terms of UQI and SSIM but with a 50% less processing time. In terms of MSE and NRMSE, the performance evaluation gives opposite results, with lower performance compared to Drago et al., Schlick et al., and Tumblin et al. but higher than Reinhard et al. Such results are in line with those obtained by Ponomarenko et al.,[39] showing that the widely used metrics such as MSE have very low correlation with human perception.

TMO	UQI	SSIM	MSE	NRMSE	Time(s)
This work (Reinhard et al. (global)[15])	0.89	0.8	79.86	0.32	5.52
Drago et al.[17]	0.54	0.58	20.91	0.1	5.43
Reinhard et al. (local)[15]	0.93	0.81	174.13	0.68	10.45
Schlick et al.[40]	0.43	0.56	5.53	0.03	5.50
Tumblin et al.[16]	0.14	0.29	4.09	0.02	5.59

Table 4 Comparison metrics derived from our test software for TMOs algorithms applied to an HDR image constructed from three exposures.

The major problem and well known of HDR technique by multiple exposures, is the difficulty to remove unwanted motion artifacts occurring during the reconstruction of the radiance map. Ghost detection and ghost removal is under research to provide better HDR video quality. The current system is limited by the bracketing spatio-temporal dissimilarities that may occur during image capture. These artifacts, can be global or local. The global ghost occurs when LDRs are misaligned during camera movement, when shooting with a hand-held camera for example. The other type of ghost comes from movement of an object in the scene during acquisition. This anomaly may render inoperative HDR imaging in some application areas. In our case, we have not implemented motion correction, but due to high framerate, the effect is only noticeable with high dynamic motions.

6 Conclusion

An HDR camera, with a complete system from capture to display, has been designed for rendering HDR content at full resolution and framerate. We show that HDR video with the original HDR technique at an high frame rate is feasible. Some effort has to be done in standardization, compression and sharing HDR datas. The multiple exposure technique can cause problems due to scene motion, but our application is not affected by this to any significant amount, as such extremely rapid scene motion does not happen in our captured scenes. This is partly due to the fact that we used a dedicated memory management core which delivers multiple videos in parallel at 60 frames per second. For extremely rapid scene motion, our HDR system may be prone to ghosting artifacts. So, we plan to study and implement onto the FPGA dedicated ghost detection techniques in order to provide a real-time ghost-free HDR live video.

References

1 R. M. Karol Myszkowski and G. Krawczyk, *High Dynamic Range Video*, Morgan et Claypool Publishers (2008).

2 A. E. Gamal, "High dynamic range image sensors," in *International Solid-State Circuits Conference*, (2002).

3 C. Jung, Y. Yang, and L. Jiao, "High dynamic range imaging on mobile devices using fusion of multiexposure images," *Optical Engineering* **52**(10), 102004–102004 (2013).

4 A. L. Gomez, S. Saravi, and E. A. Edirisinghe, "Multiexposure and multifocus image fusion with multidimensional camera shake compensation," *Optical Engineering* **52**(10), 102007–102007 (2013).

5 T. J. Park and I. K. Park, "High dynamic range image acquisition using multiple images with different apertures," *Optical Engineering* **51**(12), 127002–127002 (2012).

6 P. E. Debevec and J. Malik, "Recovering high dynamic range radiance maps from photographs," in *SIGGRAPH*, 369–378 (1997).

7 T. Mitsunaga and S. Nayar, "Radiometric Self Calibration," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, **1**, 374–380 (1999).

8 M. A. Robertson, S. Borman, and R. L. Stevenson, "Estimation-theoretic approach to dynamic range enhancement using multiple exposures," *Journal of Electronic Imaging* **12(2)**, 219–228 (2003).

9 W. S. G. Yourganov, "Acquiring high dynamic range video at video rates," tech. rep., Acquiring High Dynamic Range Video at Video Rates (2001).

10 M. D. Tocci, C. Kiser, N. Tocci, and P. Sen, "A Versatile HDR Video Production System," *ACM Transactions on Graphics (TOG) (Proceedings of SIGGRAPH 2011)* **30**(4), 9 (2011).

11 J.-H. Kim, H. Kim, and S.-J. Ko, "New visualization method for high dynamic range images in low dynamic range devices," *Optical Engineering* **50**(10), 107005–107005–7 (2011).

12 F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," tech. rep., Laboratory for Computer Science, Massachusetts Institute of Technology (2002).

13 C. Jiang Duan, MarcoBressan and GuopingQiu, "Tone-mapping high dynamic range images by novel histogram adjustment," *Pattern Recognition* **43**, 1847–1862 (2010).

14 M. W. Raanan Fattal, Dani Lischinski, "Gradient domain high dynamic range compression,"

tech. rep., School of Computer Science and Engineering The Hebrew University of Jerusalem (2002).

15 E. Reinhard, M. Stark, P. Shirley, and J. Ferwerda, "Photographic tone reproduction for digital images," *ACM Transactions on Graphics* **21**(3), 267–276 (2002).

16 S. N. Pattanaik, J. Tumblin, H. Yee, and D. P. Greenberg, "Time-dependent visual adaptation for fast realistic image display," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, *SIGGRAPH '00*, 47–54, ACM Press/Addison-Wesley Publishing Co., (New York, NY, USA) (2000).

17 T. A. F. Drago, K. Myszkowski and N. Chiba, "Adaptive logarithmic mapping for displaying high contrast scenes," *EUROGRAPHICS 2003 / P. Brunet and D. Fellner* **Volume 22 (2003), Number 3**(3) (2003).

18 M. Narwaria, M. P. Da Silva, P. Le Callet, and R. Pepion, "Tone mapping-based high-dynamic-range image compression: study of optimization criterion and perceptual quality," *Optical Engineering* **52**(10), 102008–102008 (2013).

19 A. O. Akyüz and E. Reinhard, "Perceptual evaluation of tone-reproduction operators using the cornsweet–craik–o'brien illusion," *ACM Trans. Appl. Percept.* **4**, 1:1–1:29 (2008).

20 F. Drago, W. Martens, K. Myszkowski, and H.-P. Seidel, "Perceptual evaluation of tone mapping operators with regard to similarity and preference," Research Report MPI-I-2002-4-002, Max-Planck-Institut für Informatik, Stuhlsatzenhausweg 85, 66123 Saarbrücken, Germany (2002).

21 M. Cadík, "Evaluation of hdr tone mapping methods using essential perceptual attributes," *Computers & Graphics* **32**(6), 716–719 (2008).

22 A. Akyz, "High dynamic range imaging pipeline on the gpu," *Journal of Real-Time Image Processing* **1**, 1–15 (2012).

23 S. Mann, R. Lo, K. Ovtcharov, S. Gu, D. Dai, C. Ngan, and T. Ai, "Realtime hdr (high dynamic range) video for eyetap wearable computers, fpga-based seeing aids, and glasseyes (eyetaps)," in *Electrical Computer Engineering (CCECE), 2012 25th IEEE Canadian Conference on*, 1 –6 (2012).

24 R. Ureña, P. Martinez-Cañada, J. M. Gómez-López, C. A. Morillas, and F. J. Pelayo, "Real-time tone mapping on gpu and fpga," *EURASIP J. Image and Video Processing* **2012**, 1 (2012).

25 B. Guthier, S. Kopf, and W. Effelsberg, "Optimal shutter speed sequences for real-time hdr video," in *Imaging Systems and Techniques (IST), 2012 IEEE International Conference on*, 303 –308 (2012).

26 C.-T. Chiu, T.-H. Wang, W.-M. Ke, C.-Y. Chuang, J.-S. Huang, W.-S. Wong, R.-S. Tsay, and C.-J. Wu, "Real-time tone-mapping processor with integrated photographic and gradient compression using 0.13um technology on an arm soc platform," *Journal of Signal Processing Systems* **-**, 1–15 (2010). 10.1007/s11265-010-0491-8.

27 A. K. Bachoo, "Real-time exposure fusion on a mobile computer," tech. rep., Signal Processing Research Group Optronics Sensor Systems Council for Scientific and Industrial Research (CSIR) Pretoria, South Africa (2009).

28 A. Gençtav and A. O. Akyüz, "Evaluation of radiometric camera response recovery methods," in *SIGGRAPH Asia 2011 Posters*, *SA '11*, 15:1–15:1, ACM, (New York, NY, USA) (2011).

29 M. Ali and S. Mann, "Comparametric image compositing: Computationally efficient high

dynamic range imaging," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, 913 –916 (2012).

30 B. Guthier, S. Kopf, and W. Effelsberg, "A real-time system for capturing hdr videos," in *Proceedings of the 20th ACM international conference on Multimedia*, *MM '12*, 1473–1476, ACM, (New York, NY, USA) (2012).

31 F. V. R. Tom Mertens, Jan Kautz, "Exposure fusion," tech. rep., Hasselt University EDM transationale Universiteit Limburg Belgium, University College London UK (2007).

32 A. A. Goshtasby, "Fusion of multi-exposure images," *Image and Vision Computing* **23**, 611–618 (2005).

33 N. Gelfand, A. Adams, S. H. Park, and K. Pulli, "Multi-exposure imaging on mobile devices," in *Proceedings of the international conference on Multimedia*, *MM '10*, 823–826, ACM, (New York, NY, USA) (2010).

34 J. Detrey, *Arithmétiques réelles sur FPGA : virgule fixe, virgule flottante et système logarithmique*. PhD thesis, École Normale Supérieure de Lyon, Lyon, France (2007).

35 A. Yoshida, V. Blanz, K. Myszkowski, and H. peter Seidel, "Perceptual evaluation of tone mapping operators with real-world scenes," in *Human Vision & Electronic Imaging X, SPIE*, 192–203, SPIE (2005).

36 Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on* **13**(4), 600–612 (2004).

37 Z. Wang and A. C. Bovik, "A Universal image Quality Index," *Signal Processing Letters, IEEE* **9**(3), 81–84 (2002).

38 D. Rouse and S. Hemami, "Understanding and simplifying the structural similarity metric," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, 1188–1191 (2008).

39 N. Ponomarenko, F. Battisti, K. Egiazarian, J. Astola, and V. Lukin, "Metrics performance comparison for color image database," in *Proc. International Workshop on Video Processing and Quality Metrics*, (2009).

40 C. Schlick, "Quantization techniques for visualization of high dynamic range pictures," in *Photorealistic Rendering Techniques*, G. Sakas, S. Mller, and P. Shirley, Eds., *Focus on Computer Graphics*, 7–20, Springer Berlin Heidelberg (1995).

List of Figures

4 Memory Management Core. Performing three parallel streaming videos with low (I_1), middle (I_2) and high (I_3) exposure times. The delayed HDR row output is shown after HDR and tone mapping computations (related to Section 4.2 and 4.3).

5 HDR creating and tone mapping hardware pipeline using three different pixel streams. Frame enable is active when a new HDR frame is coming. It is important to note that we use IEEE754 32-bit floating-point arithmetic operators.

6 Results of the complete system. Our Multiple Exposure Control can select the three proper exposures, and the specific memory management core permits us to display the 3 bracketed images. The HDR image is in the right of each image set.

7 Output video (MPEG, 3 MB).

8 The three low dynamic range images used for quality comparison.

List of Tables

1 Summary of the main embedded real-time architectures dedicated to HDR.

2 Resource comparison of fixed and floating point arithmetic operators on Virtex 6

3 Summary of hardware implementation results on the Virtex-6 platform.

4 Comparison metrics derived from our test software for TMOs algorithms applied to an HDR image constructed from three exposures.